Software Reliability Analysis Tools

Joel Henry, Ph.D.

University of Montana

Outline

- Background
- MATLAB Automated Testing Tool
- Graphical Input Specification Tool
- Real-Time Analysis Testing Tool
- Status
- Conclusion

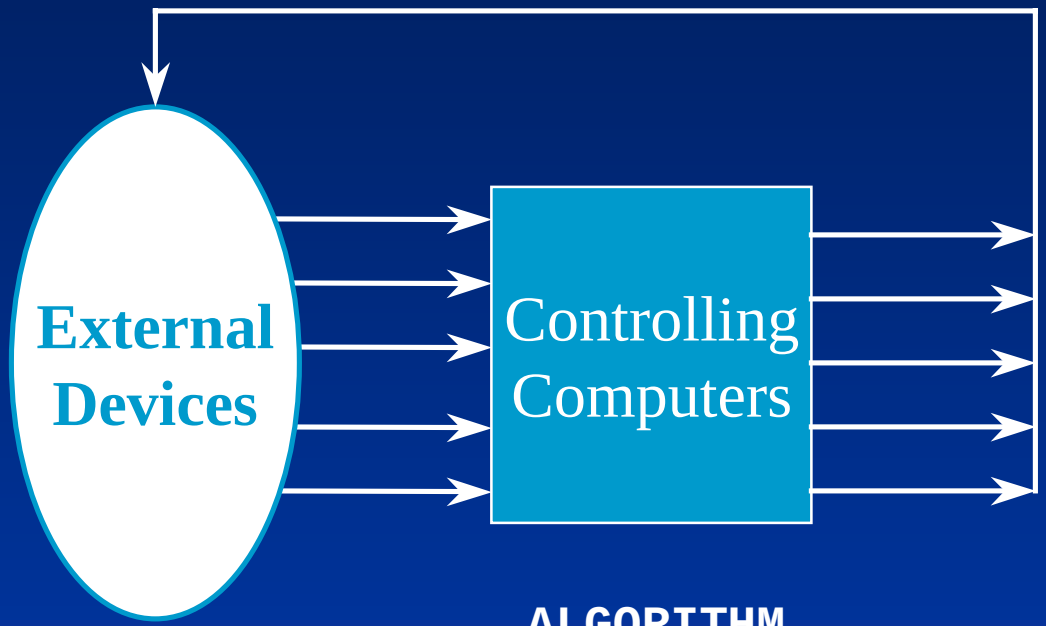
System Development

Example: Wind Tunnel Software

- Controls devices that control:
  - Wind generator
  - Model support
  - Tunnel atmosphere
- Emphasizes reliability and safety
- Utilizes multiple development strategies
- Based on simple structure

System Development

Simple Structure



ALGORITHM

Sample Inputs
Run Software
Update Outputs

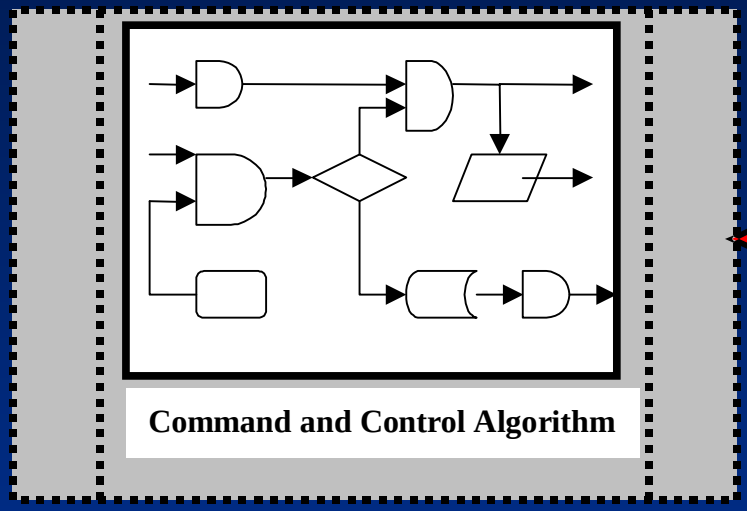
Testing Problems

- Size
  - Input variables – sampled over time
  - Outputs variables – produced over time
  - Sample time – variable or set frequency
- Requirements
  - Input file/matrix
  - Output file/matrix
  - Analysis tools
- Domain determinants
  - Input variable – minimum, maximum, and accuracy
  - Output variable – minimum, maximum, and accuracy
- Test requirements
  - Input file/matrix with all possible values for input
  - Output file/matrix much more complex problem

Solution Approach
Overview

- Automation to:
  - Generate large input matrices/files
  - Perform simulation and/or test auto-generated code
  - Analyze output matrices/files
- Methods to:
  - Evaluate domain coverage
  - Aid debugging
  - Evaluate results

Solution Approach



MATLAB/Simulink Environment

Simulation

Suite of testing tools
- Generate Tests
- Simulate Model
- Test Auto-code
- Detect Faults
- Evaluate Results

Verification and Validation Methodology

Tests

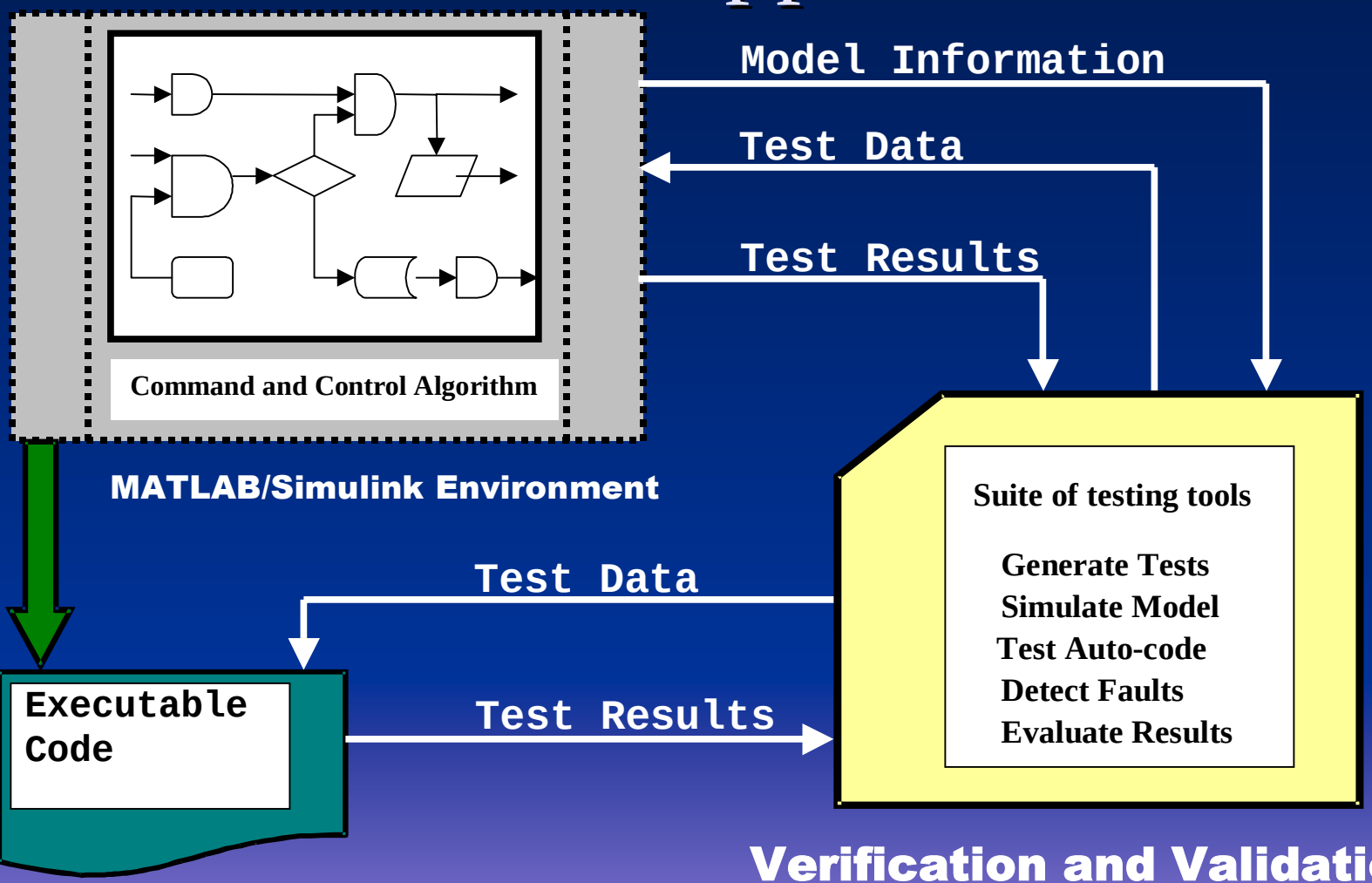
Autocode

Source Code

Compile/Link

Executable Code

Solution Approach



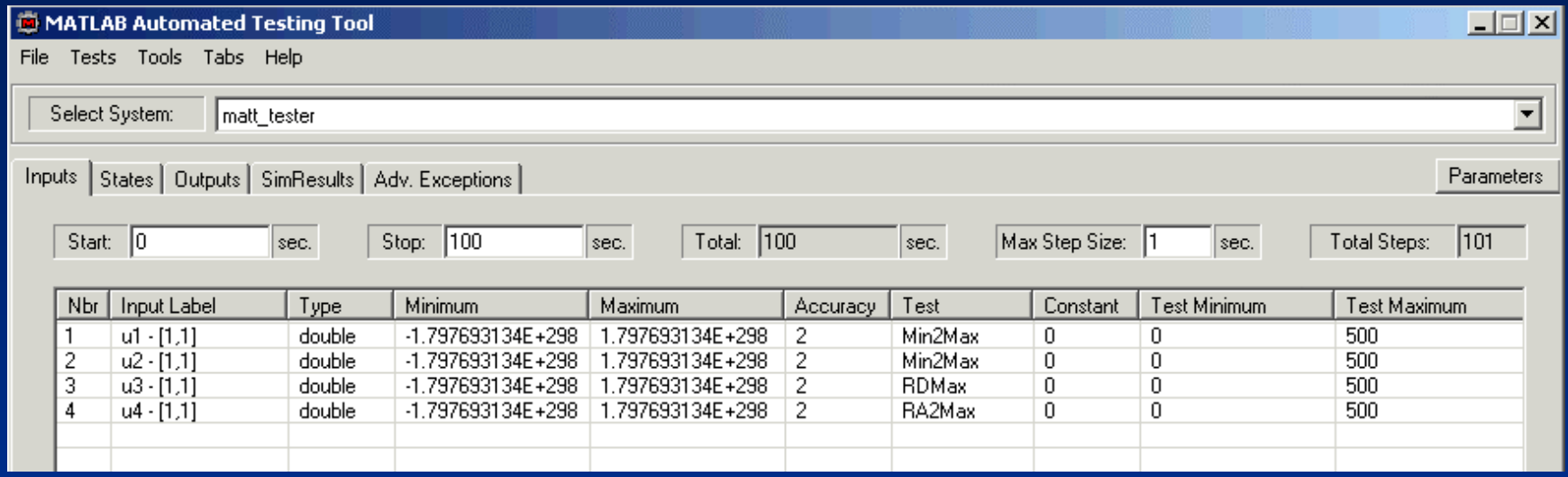
Verification and Validation Methodology

MATLAB Automated Testing Tool

- Creates test data
- Executes simulation or test
- Captures results
- Detects exceptions
- Saves test data or complete test results

MATT – Test Setup

Inputs

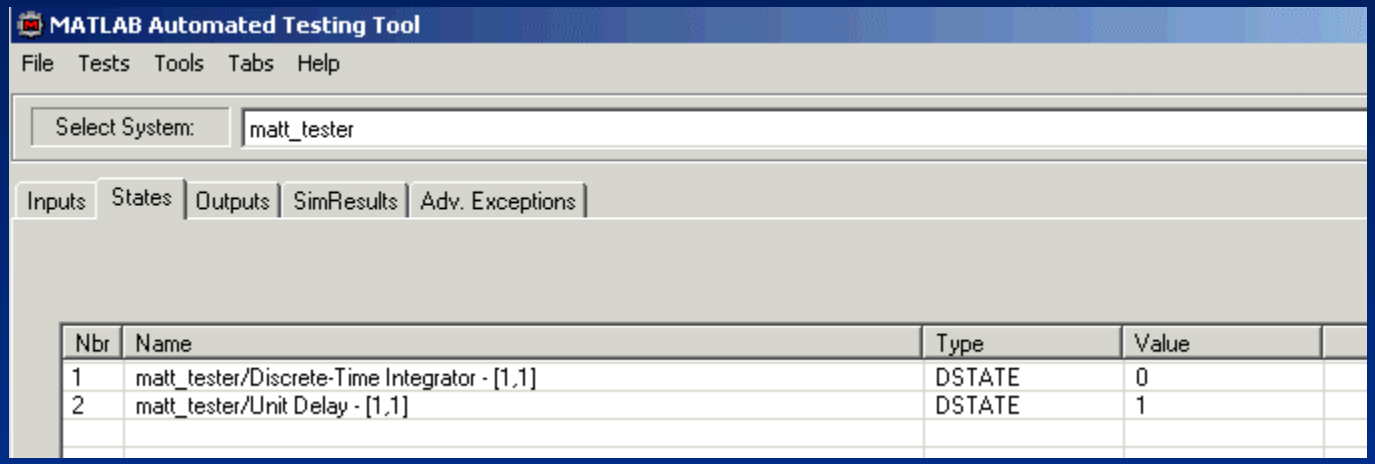


Nbr	Input Label	Type	Minimum	Maximum	Accuracy	Test	Constant	Test Minimum	Test Maximum
1	u1 - [1,1]	double	-1.797693134E+298	1.797693134E+298	2	Min2Max	0	0	500
2	u2 - [1,1]	double	-1.797693134E+298	1.797693134E+298	2	Min2Max	0	0	500
3	u3 - [1,1]	double	-1.797693134E+298	1.797693134E+298	2	RDMax	0	0	500
4	u4 - [1,1]	double	-1.797693134E+298	1.797693134E+298	2	RA2Max	0	0	500

- Set test running time
- Select and configure tests using 31 test types
- Now supports non-scalar inputs*

MATT – Test Setup

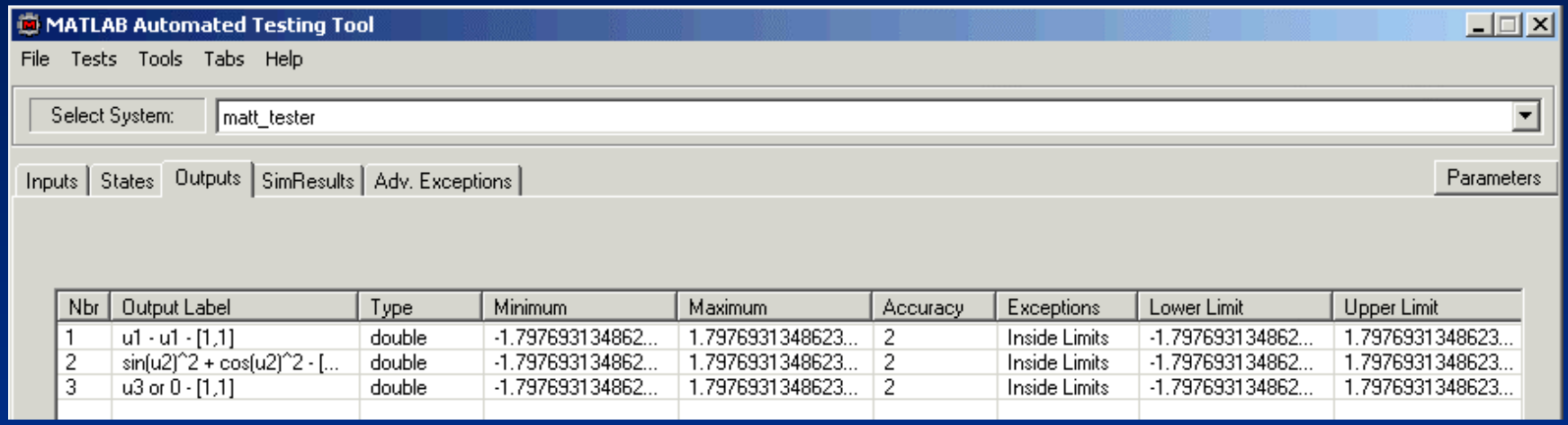
States



- Assign Starting values for state blocks*

MATT – Test Setup

Outputs



Nbr	Output Label	Type	Minimum	Maximum	Accuracy	Exceptions	Lower Limit	Upper Limit
1	u1 - u1 - [1,1]	double	-1.797693134862...	1.7976931348623...	2	Inside Limits	-1.797693134862...	1.7976931348623...
2	sin(u2)^2 + cos(u2)^2 - [...	double	-1.797693134862...	1.7976931348623...	2	Inside Limits	-1.797693134862...	1.7976931348623...
3	u3 or 0 - [1,1]	double	-1.797693134862...	1.7976931348623...	2	Inside Limits	-1.797693134862...	1.7976931348623...

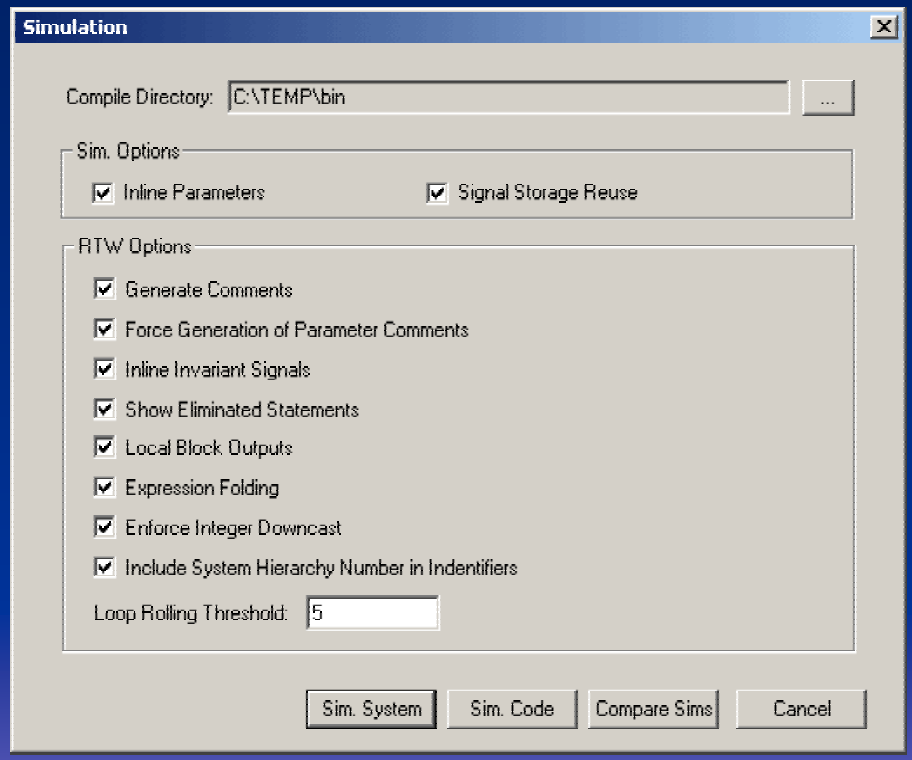
- Set output accuracy and exception ranges
- Now supports non-scalar outputs*

MATT – Test Setup

New Exception Types

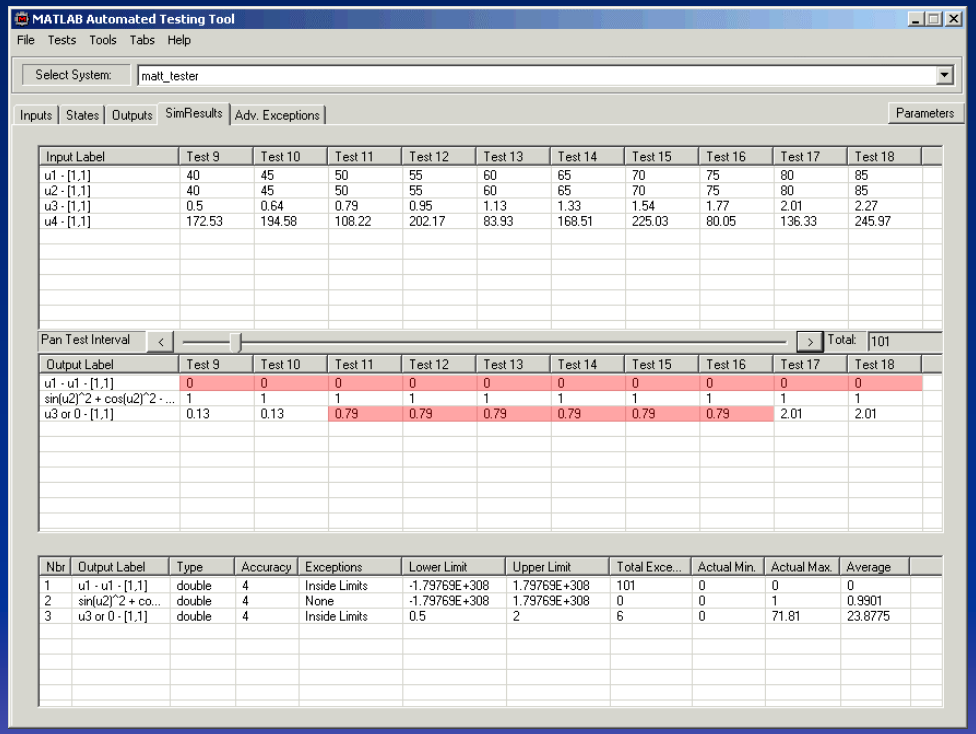
- Percent change*
  - Allows exception detection if the output value changes more than a specified percent over a specified number of steps
- Absolute change*
  - Allows exception detection if the output value changes more than a specified amount over a specified number of steps

MATT – Simulate



- Set up simulation and code generation options
- Simulate:
  - Model
  - Auto-generated Code
  - Compare both

MATT – View Results



- View input values for every time-step
- View output values for every time-step
  - Steps causing exceptions are highlighted red*
- View/edit output exception info and settings

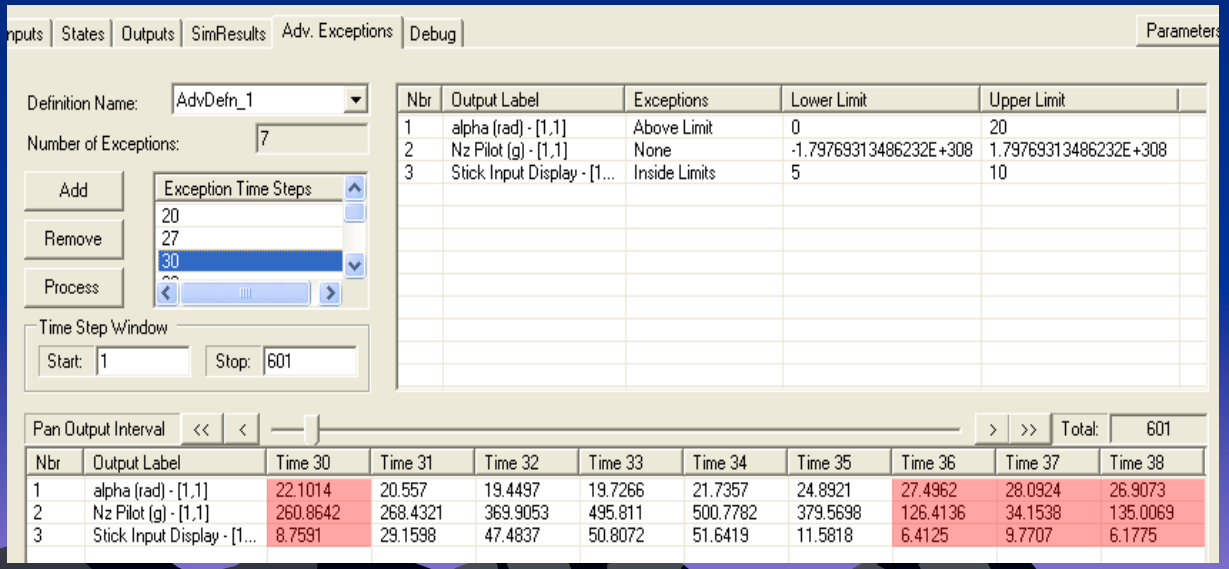
MATT – Advanced Exceptions*

- Advanced Exceptions allow exception detection based on multiple exception criteria
  - A combination of output ports
  - Disjoint ranges
    - Create separate A.E definition for each range that needs to be tested
    - Do a combinational process based on all the A.E definitions
  - Overall system reliability
    - Create multiple A.E definitions based on system specifications
    - Perform a combinational process based on all the defined Advanced Exceptions

MATT – Advanced Exceptions

Combination of output ports

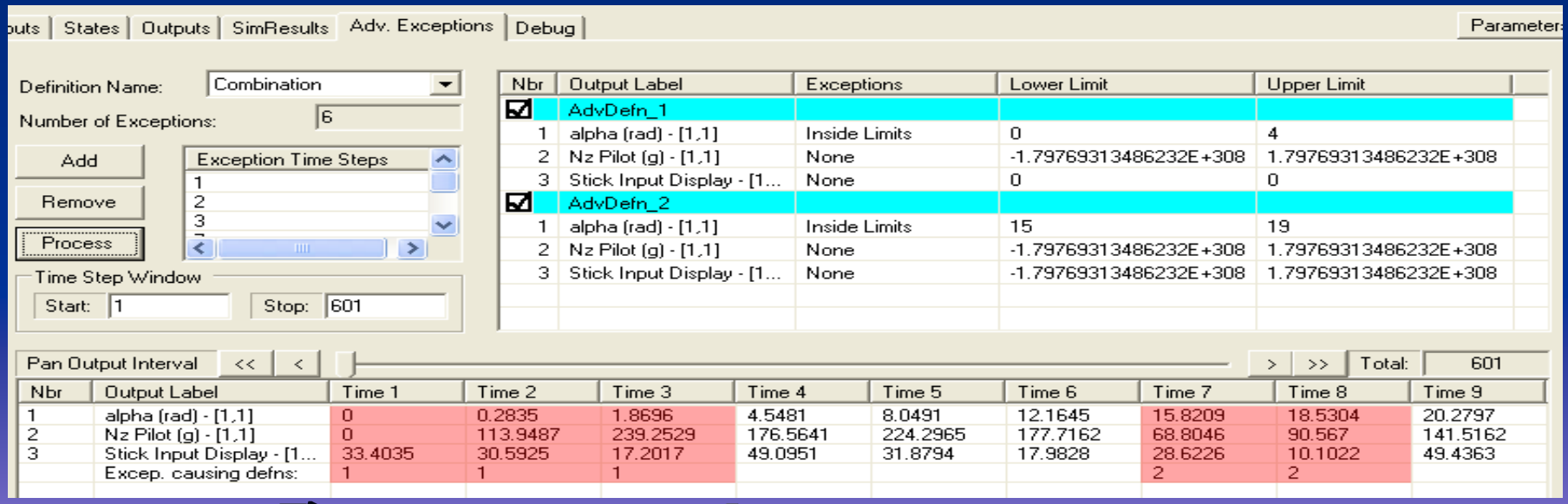
- In the example shown below, an exception is said to have occurred when:
  - Stick output is between 5 and 10
  - Alpha output is greater than 20 rads
- After clicking the *process* button, one discovers:
  - 7 exceptions occurred from the given exception definition
  - The time steps where the 7 exceptions occurred
  - The output values that produced the 7 exceptions



MATT – Advanced Exceptions

Disjoint Ranges

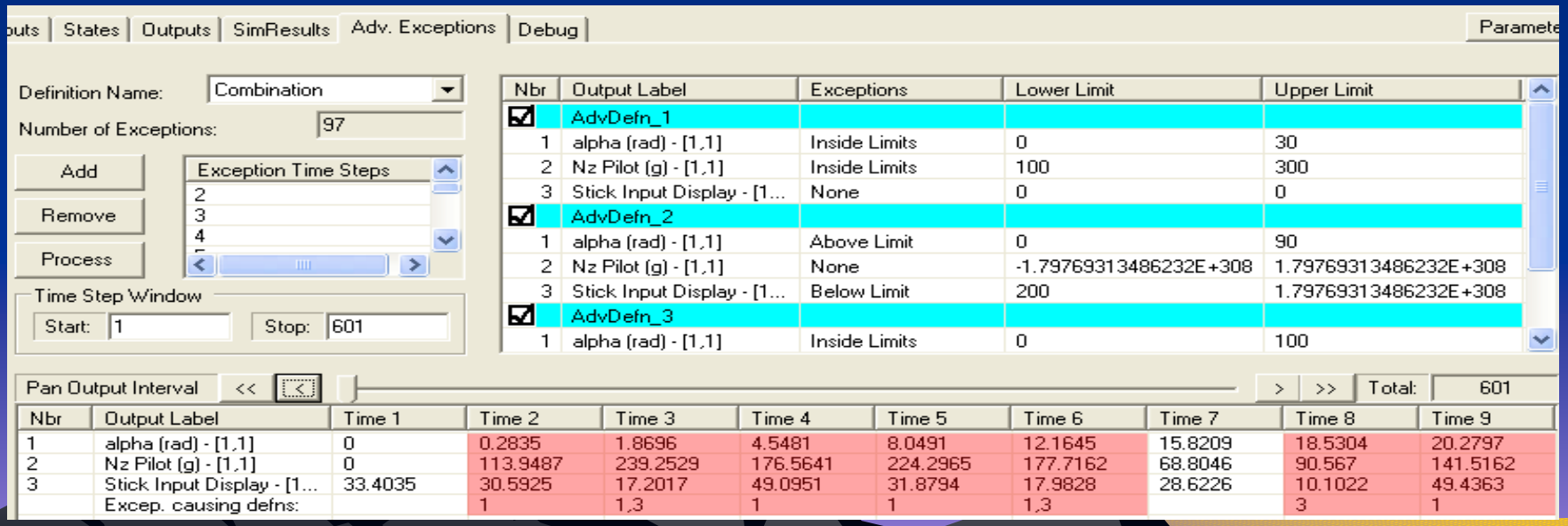
- In this example, two separate exception definitions are specified:
  - (1) Alpha producing values between 0 - 4 rads
  - (2) Alpha producing values between 15 – 19 rads
- These two exception definitions are combined, and exception results are calculated in aggregate.



MATT – Advanced Exceptions

Overall System Reliability

- **By combining multiple A.E. definitions, systems can be stress-tested for accuracy and reliability.**



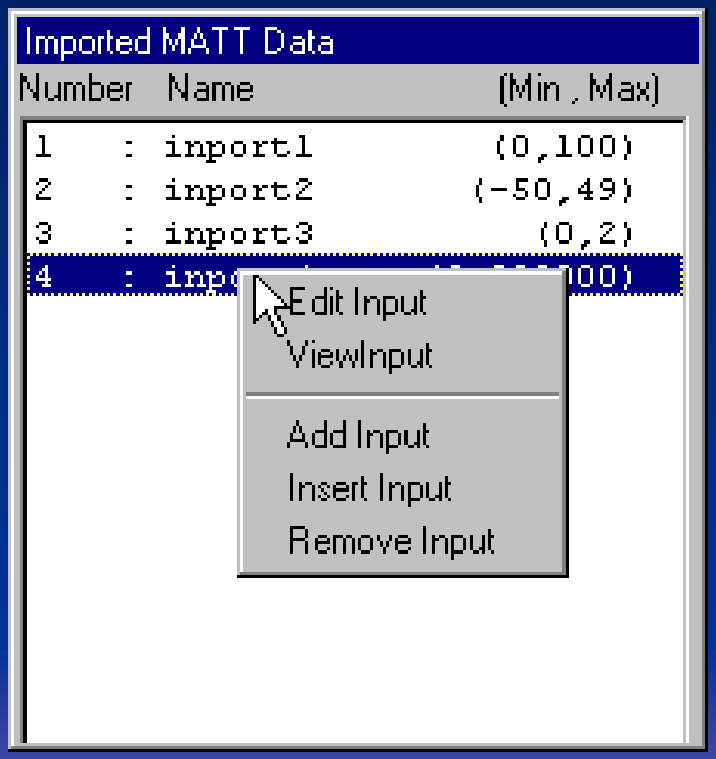
MATT - Storage

- Save complete test setup for future use
  - Input, State, and Output settings
  - Advanced Exception Setting
  - Input Matrix
  - Output Matrix
- Import and export custom input matrices
  - Comma delimited files (*.csv)
  - MATLAB data files (*.mat)

GIST

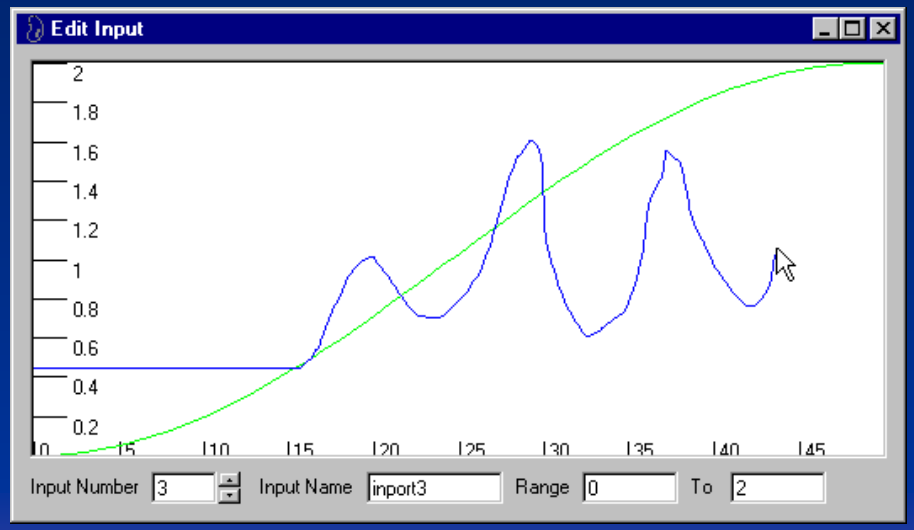
- Reads MATT test data files
- Allows freehand specification of test data
- Saves new MATT test data files

GIST – Manage Input Sets



- Add/Remove inputs for altered models
- View graphs of any input
- Edit any input

GIST – Edit Inputs

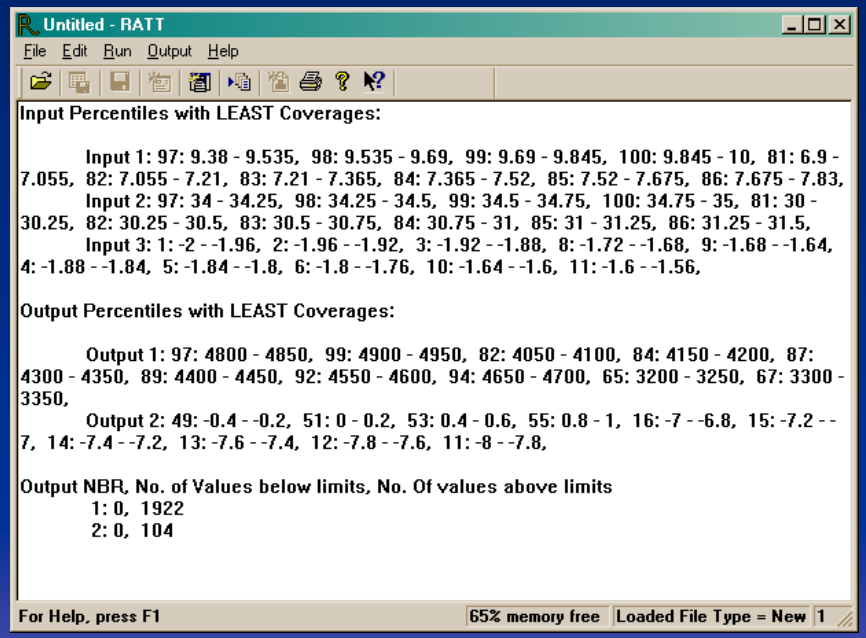


- Use simple click and drag to create new input values
- View original input values as drawing

RATT

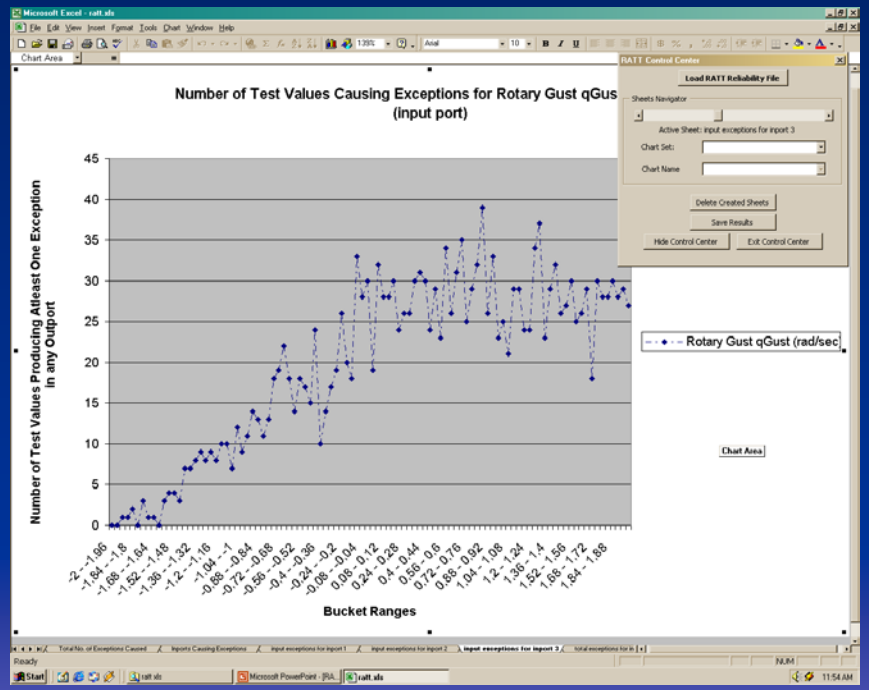
- Reads MATT test result files
- Calculates reliability, probabilities, and completeness measures
- Supports analysis of multiple test files (test suite)
- Exports to MS Excel and populates analysis charts and graphs

RATT - Analysis



- Load several MATT test files at once
- Perform analysis
  - Input coverage
  - Output coverage
  - MTTF
  - Probability of failure
  - Etc
- View simple results in RATT

RATT – Excel® Tool



- Load RATT files into Excel®
- View graphs showing coverage
- View coverage on multiple inputs/exceptions at the same time

Current Status

- Windows and UNIX versions of MATT ready
- Windows versions of GIST and RATT ready
- Working with STEREO project
  - Testing MATLAB/Simulink® models using MATT

Upcoming Functionality

- MATT
  - View intermediate state values from simulation
  - Create and use custom test-types
  - Load models inside MATT
    - Allows MATT to launch outside of MATLAB®
- GIST
  - Being integrated into MATT for more seamless use
- RATT
  - Being revised to accommodate Advanced Exceptions

Conclusion – Testing Enhanced

- Using software tools to aid testing allows
  - More tests to be run
  - Each test to be run more quickly
  - Test results to be easily stored and referenced

Conclusion – Tools Enhanced

- Improved MATT capabilities allow
  - Systems to start at any state with ease
  - More advanced exception catching

Questions and Contact Info

- Joel Henry
  - henryj@cs.umt.edu
  - MATT, RATT, and GIST
  - **http://www.cs.umt.edu/RTSL/matt/**
- MATLAB and Simulink users